

Article

Test-Time Adaptation for Personal Voice Activity Detection: VAD-Gated Test-Time Training and Speaker Embedding Adaptation

Tai-You Chen, Chien-Chia Chiu, Jung-Shan Lin and Jeih-Weih Hung * 

Department of Electrical Engineering, National Chi Nan University, Nantou County 545, Taiwan; s112323047@ncnu.edu.tw (T.-Y.C.); s112323055@mail1.ncnu.edu.tw (C.-C.C.); jslin@ncnu.edu.tw (J.-S.L.)

* Correspondence: jwhung@ncnu.edu.tw

Abstract

Personal voice activity detection (PVAD) identifies whether each detected speech frame originates from a designated target speaker. Modern PVAD systems are typically trained offline and then deployed with frozen model parameters and a fixed, pre-enrolled speaker embedding, leaving them unable to adapt to distribution shifts at inference time such as unseen acoustic environments, changing speaking styles, or mismatches between enrollment and test conditions. Test-time training (TTT) and test-time adaptation have shown promise in language, vision, and several speech tasks, yet their behavior on PVAD has not been studied. In this work, we present an empirical study of two complementary test-time adaptation mechanisms built on top of the recently proposed FDE-Mamba backbone. The first is a VAD-gated TTT adapter, which instantiates the TTT-Linear formulation within the personalization pathway and augments it with a VAD-probability gate and exponential moving-average stabilization, adapting an internal weight matrix on the speaker-conditioned feature stream of each test utterance. The second is TEA (Test-time Embedding Adaptation), a scheme that keeps all model parameters frozen and instead adapts the target speaker d-vector itself via self-supervised objectives at inference time, directly targeting enrollment–test mismatch. We evaluate both mechanisms on the LibriSpeech PVAD benchmark across two backbones (LSTM-based FDE-RNN and Mamba-based FDE-Mamba), reporting category-wise average precision, mean average precision (mAP), accuracy, recall, precision, and real-time factor. We further isolate the effect of a post-hoc Gaussian smoothing step and report that, of all the components we examine, this task-agnostic smoothing accounts for the largest single accuracy gain on the FDE-Mamba backbone (accuracy 89.87% → 90.47%); the test-time adaptation mechanisms contribute a separate, smaller gain that is concentrated on speaker-discrimination metrics (mAP, precision) rather than on accuracy. Overall, the proposed test-time adaptation yields a consistent but modest improvement over the FDE-Mamba baseline (mAP 0.9605 → 0.9641, precision 0.881 → 0.899), while slightly reducing recall and increasing inference cost when TEA is enabled. Through ablation studies, we quantify the independent and combined contribution of each component and characterize the recall–precision trade-off introduced by adaptation. These findings, together with a discussion of their limitations and cost–benefit profile, provide a measured baseline and design insights for future work on adaptive PVAD, particularly under stronger acoustic and enrollment mismatches than those captured by the LibriSpeech protocol.



Academic Editor: Honarvar Shakibaei Asli

Received: 13 June 2026

Revised: 7 July 2026

Accepted: 13 July 2026

Published: 15 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](#)

[Attribution \(CC BY\)](#) license.

Keywords: personal voice activity detection; voice activity detection; test-time training; test-time adaptation; test-time embedding adaptation; Mamba; selective state space model; entropy minimization

1. Introduction

Voice-based human–machine interaction has become a central interface in modern smart devices, creating sustained demand for robust and efficient speech processing front-ends. Voice Activity Detection (VAD), which classifies incoming audio frames as speech or non-speech, is a fundamental preprocessing stage for downstream systems such as automatic speech recognition and voice assistants [1,2]. Beyond filtering out non-speech regions, VAD also plays an important role in reducing power consumption and managing computational resources on resource-constrained devices.

Conventional VAD, however, cannot determine whether detected speech belongs to a specific target speaker—a limitation that becomes critical in multi-speaker environments. Personal Voice Activity Detection (PVAD) was introduced to address this gap, extending the binary VAD formulation into a three-way classification over target speaker speech (tss), non-target speaker speech (ntss), and non-speech (ns) [3]. By conditioning on a target speaker embedding derived from a speaker verification model, PVAD systems can respond selectively to a designated speaker while suppressing competing voices, making them well suited to personal voice assistant applications. A substantial body of work has refined how speaker embeddings are fused with acoustic features [4–7], culminating most recently in the Flexible Dynamic Encoder RNN (FDE-RNN) [8] and its selective state-space successor, FDE-Mamba [9], which together represent the current state of the art on this benchmark.

Despite these architectural advances, essentially all PVAD systems share a common deployment assumption: once trained, the model is frozen and operates with a fixed, pre-enrolled speaker embedding. The model parameters do not change in response to the audio actually encountered at inference time, and the speaker representation is computed once from enrollment speech and then held constant.

This static design is particularly consequential for PVAD, more so than for ordinary VAD. Because PVAD makes a *speaker-conditioned* decision, its accuracy depends not only on how well the model separates speech from non-speech, but also on how faithfully the fixed enrollment embedding represents the target speaker as they actually sound in the test utterance, and on whether the acoustic conditions of that utterance resemble those seen during training. Both assumptions are fragile in practice. The acoustic environment at test time—noise, reverberation, recording channel, and device—routinely differs from the training distribution, shifting the feature statistics on which the speaker-discrimination decision is based. A speaker’s voice is itself non-stationary: speaking rate, vocal effort, emotional state, and health vary across sessions, so an embedding enrolled on one occasion may be a poor match for the same speaker on another. The enrollment utterance may also be short, noisy, or recorded under conditions unlike the test signal, yielding an embedding that is biased from the outset. In each of these cases the mismatch falls precisely on the speaker-conditioning pathway that distinguishes PVAD from VAD, and a model that cannot adapt at inference time has no recourse: it must rely entirely on the generalization acquired during offline training, even when the test utterance in front of it carries information that could be used to correct the mismatch. This observation motivates asking whether PVAD can instead adapt to each test utterance as it is processed.

A growing line of research addresses exactly this problem through *test-time adaptation*, a family of methods in which a trained model adjusts itself to the test distribution using

only the unlabeled test signal, without any ground-truth labels. The central difficulty is that adaptation must be driven by a signal that is available at inference, and different methods differ in what self-supervised signal they use and what they update. Entropy minimization, exemplified by TENT [10], updates a small set of model parameters—typically the affine parameters of normalization layers—so as to reduce the entropy of the model’s own predictions, on the premise that confident predictions tend to be correct ones; it has been applied to speech recognition and to speaker adaptation for lip reading [11], usually resetting the adapted parameters after each utterance. Test-Time Training (TTT) takes a different route. Introduced by Sun et al. [12] to combat train–test distribution shift, it pairs the main task with an auxiliary self-supervised task during training; at inference, the self-supervised task is used to fine-tune the model on each test input before the main prediction is made, so that the representation is specialized to the local statistics of that input. A recent reformulation [13] casts TTT as a sequence-modeling layer whose hidden state is *itself* a small machine-learning model: the hidden state is a weight matrix $\mathbf{W}$, and the update rule is a step of gradient descent on a self-supervised reconstruction loss applied at every position, so that $\mathbf{W}$ keeps learning even as it processes a test sequence. This view is appealing for a streaming, speaker-conditioned task such as PVAD for two reasons. First, the self-supervised reconstruction signal is computed directly from the incoming features, so adaptation is driven by the very utterance whose speaker decision is at stake—exactly the information a frozen model discards. Second, because the adapted state is a weight matrix updated online, it can absorb a distribution shift expressed jointly across many feature dimensions, rather than requiring the shift to be attributed to any single pre-specified factor. TTT has shown benefits in language modeling [13], computer vision [12], and more recently in speech tasks such as speech enhancement [14] and speech editing, where adapting to unseen noise or speaker conditions at inference improves output quality.

To date, however, the behavior of these adaptation mechanisms on PVAD has not been investigated. PVAD differs from previously studied tasks in two respects that make this an open question rather than a foregone conclusion: it is an inherently speaker-dependent task, and its decision depends jointly on the acoustic signal and an externally supplied speaker representation. It is therefore not obvious *a priori* whether test-time adaptation helps, how large any benefit is, or—given that PVAD has two adaptable surfaces, the internal features and the speaker embedding—which part of the system should adapt. These are empirical questions specific to PVAD, and answering them is the purpose of this study.

In this work we conduct an empirical study of test-time adaptation for PVAD, built on the FDE-Mamba backbone. We investigate two complementary mechanisms. The first, a *VAD-gated TTT adapter*, instantiates the TTT-Linear formulation [13] within the personalization pathway, adapting an internal weight matrix on the speaker-conditioned feature stream of each utterance. Crucially, this adapter operates on features *after* speaker-embedding fusion, so the distribution shift it absorbs reflects the combined influence of acoustic conditions, speaking style, and speaker characteristics on the test signal, rather than any single factor in isolation. We tailor the adapter to PVAD with a VAD-probability gate, which restricts adaptation to frames classified as speech, and an exponential moving-average update, which stabilizes the weight trajectory during inference. The second mechanism, *TEA* (Test-time Embedding Adaptation), performs test-time adaptation of the target speaker embedding: with all model parameters frozen, it updates the d-vector itself through self-supervised objectives, directly targeting the enrollment–test mismatch. Throughout, we treat the study as a measurement exercise: we quantify each component’s contribution independently and in combination, report a recall–precision trade-off that adaptation introduces, and, importantly, separately isolate the effect of a post hoc Gaussian smoothing step whose contribution turns out to be the largest single accuracy gain we observe on the

FDE-Mamba backbone but which is unrelated to the adaptation mechanisms and should not be attributed to them.

The contributions of this paper are as follows:

- To the best of our knowledge, this is the first work to apply Test-Time Training to PVAD. We adapt the TTT-Linear formulation with a VAD-probability gate and an exponential moving-average update tailored to the speech-dependent, streaming nature of PVAD, and we clarify that the adapter absorbs a composite distribution shift over the speaker-conditioned feature stream rather than adapting the speaker embedding alone.
- We introduce TEA (Test-time Embedding Adaptation), a test-time speaker embedding adaptation scheme for PVAD. While entropy-minimization-based test-time adaptation has been studied for other speech tasks, we are, to our knowledge, the first to apply it to PVAD and, distinctively, to adapt the target speaker d-vector itself rather than the model's internal parameters.
- We present a systematic empirical study that quantifies the independent and combined effect of each mechanism, analyzes the recall–precision trade-off they induce, and isolates the contribution of post-hoc Gaussian smoothing—reporting the finding, which the ablation makes explicit, that this task-agnostic smoothing step is the largest single contributor to accuracy on our backbone and that the adaptation mechanisms contribute a separate, smaller improvement concentrated on speaker-discrimination metrics. We further show that the observed behavior holds across both an LSTM-based (FDE-RNN) and a Mamba-based (FDE-Mamba) backbone. We report the gains as consistent but modest, and discuss the limitations that this implies for adaptive PVAD.

The remainder of this paper is organized as follows. Section 2 reviews PVAD and test-time adaptation. Section 3 describes the FDE-Mamba backbone and the two adaptation mechanisms. Section 4 details the experimental setup, and Section 5 reports the main results against the FDE-Mamba backbone, an ablation of each component, the computational cost, a generalization check on the FDE-RNN backbone, a comparison against TENT as an existing TTA baseline, a discussion of limitations, and a controlled distribution-shift analysis that characterizes how the observed gains scale with the amount of mismatch present at inference time. Section 6 concludes.

2. Related Work

2.1. Personal Voice Activity Detection

Personal Voice Activity Detection extends conventional VAD by incorporating target speaker information, enabling the system to distinguish target speaker speech (tss) from non-target speaker speech (ntss) and non-speech (ns). The seminal work of Ding et al. [3] introduced a speaker-conditioned framework that fuses a d-vector speaker embedding [15] with acoustic features for ternary frame-level classification. Personal VAD 2.0 [4] subsequently reduced model complexity for on-device speech recognition while maintaining competitive performance.

Later work explored richer strategies for integrating speaker and acoustic information. AS-pVAD [6] proposed a frame-wise attentive score loss to sharpen target-speaker detection; COIN-AT-PVAD [16] introduced conditional intermediate attention within a stacked encoder–decoder; Makishima et al. [17] studied enrollment-free training to remove the dependency on explicit enrollment utterances; and SE-PVAD [18] replaced log-Mel features with a learnable extractor for improved input discrimination. Most recently, FDE-RNN [8] proposed a flexible dynamic architecture that decouples VAD and PVAD into a shared Dynamic Encoder RNN backbone and a detachable personalization module, using a gating-based skipping mechanism to bypass redundant computation on non-speech frames.

FDE-Mamba [9] subsequently replaced the LSTM components of FDE-RNN with selective state-space (Mamba) [19] blocks, improving accuracy and reducing the real-time factor while preserving the dynamic gating design.

A common thread across these systems is a doubly static deployment assumption: the speaker embedding is computed once from enrollment speech and then held fixed, *and* the model itself is frozen after training. The first assumption leaves the system unable to correct a mismatch between the enrollment embedding and the target speaker as they actually sound at test time; the second leaves it unable to adapt to the acoustic conditions of the test utterance—noise, reverberation, channel, or speaking style—when these differ from the training distribution. Robustness to such adverse conditions has been pursued primarily through training-time interventions, such as multi-condition training or self-supervised pretraining [20], which must anticipate the test distribution in advance. The complementary question of whether a trained PVAD model can instead be *adapted at inference time*, using the test signal itself—adapting both its internal representation to the acoustic conditions of each utterance and its speaker representation to the target—has received little attention, and it is this question that the present study addresses.

2.2. Test-Time Adaptation and Test-Time Training

Test-time adaptation denotes a family of methods that adjust a trained model to the test distribution using only the unlabeled test data encountered at inference, without ground-truth labels. Methods in this family are distinguished by two design choices: the self-supervised signal that drives adaptation, and the subset of quantities that are updated. We organize the relevant literature along these axes.

TTT as a distribution-shift remedy. Test-Time Training was introduced by Sun et al. [12] to combat train–test distribution shift. The model has a Y-shaped architecture: a shared feature encoder with parameters θ_e feeds two branches—a main-task head θ_m and a self-supervised auxiliary head θ_s . During training, all three parameter sets are optimized jointly. At inference, for each individual test input the auxiliary head provides a self-supervised loss $\mathcal{L}_s$ (computed without labels), and a few gradient steps on $\mathcal{L}_s$ update only the shared encoder θ_e —the two heads stay fixed—before the main prediction is read out. Because the adaptation is performed per input and then discarded, TTT treats each test example as a one-sample learning problem, trading additional inference computation for robustness to shifts unseen during training. Subsequent work extended this principle to a range of domains, including 3D reconstruction and medical imaging, and explored which subset of parameters to adapt—from normalization statistics to low-rank or bias-only updates—in order to control the cost and stability of the inner-loop update.

TTT as a sequence-modeling layer. A more recent line of work [13] reinterprets TTT not as an add-on adaptation procedure but as a sequence-modeling layer in its own right. The key idea is to make the layer’s hidden state a small machine-learning model whose parameters are a weight matrix $\mathbf{W}$, and to define the recurrence as a step of self-supervised learning: at each position t , given an input feature (denoted $\tilde{\mathbf{F}}_t$ to match the notation we use later for the speaker-conditioned stream), $\mathbf{W}$ is updated by gradient descent on a reconstruction loss between a corrupted and a clean projection of $\tilde{\mathbf{F}}_t$, and the layer output $\mathbf{z}_t$ is $\tilde{\mathbf{F}}_t$ mapped through the updated $\mathbf{W}$. The authors instantiate this as TTT-Linear and TTT-MLP, where $\mathbf{W}$ parameterizes a linear map and a two-layer network, respectively. Viewed this way, the hidden state “learns” on the test sequence with linear inference complexity, and the construction has been shown to compete with attention on long-context language modeling and to transfer to video generation and image classification. Our TTT adapter builds on this formulation; we describe the modifications it requires for PVAD in Section 3.3.

Figure 1 contrasts the two formulations; symbols are defined here and, for the components specific to our model, in Section 3.

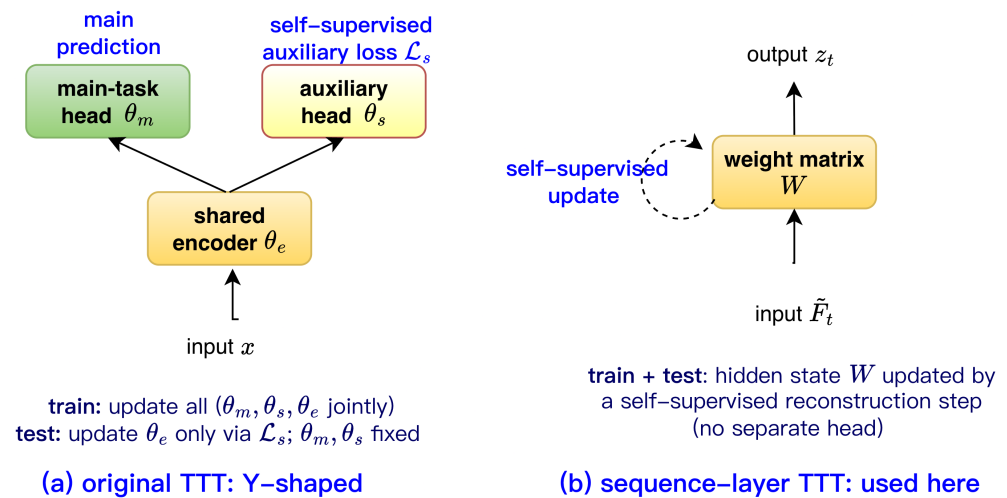


Figure 1. Two formulations of Test-Time Training (TTT). (a) *Original TTT* [12]: a Y-shaped architecture comprising a shared encoder θ_e , a main-task head θ_m , and a self-supervised auxiliary head θ_s . At training time, all three parameter sets are updated jointly. At test time, the auxiliary loss $\mathcal{L}_s$ is used to update *only* the shared encoder θ_e ; both heads remain fixed. (b) *Sequence-layer TTT* [13] (used in this work): the hidden state is a weight matrix W that is updated by a self-supervised reconstruction step at every position, during both training and inference, with no separate auxiliary head. Components shown in blue are updated by the self-supervised signal.

Entropy minimization and parameter-level adaptation. A parallel and widely used approach is entropy minimization. TENT (*test entropy* minimization) [10] adapts the affine parameters of normalization layers at test time so as to minimize the entropy of the model's predictions, on the premise that lower-entropy predictions are more reliable. Later methods refine this with confidence-based sample filtering, sequence-level objectives suited to structured outputs, and moving-average stabilization for continual adaptation [21]. What these methods share is that the adapted quantity is a set of *model parameters*—most often normalization or affine weights—rather than any input to the model.

Test-time adaptation in speech. Within speech processing, both threads have begun to appear. Dumpala et al. [22] introduced TTT for speech, reporting improvements on several downstream tasks under shifts in speaking style, gender, and age. TTT has since been applied to speech enhancement [14], where a self-supervised auxiliary task lets the model adapt to unseen noise at inference, and to speech editing. On the entropy-minimization side, test-time adaptation has been used for speech recognition and for speaker adaptation in lip reading [11], typically updating a subset of model parameters and resetting them after each utterance.

Two gaps follow from this survey and motivate our study. First, although TTT has been applied to several speech tasks, it has not been applied to PVAD; PVAD's speaker-dependent, dual-input nature—a decision jointly determined by the acoustic stream and an external speaker embedding—makes its response to TTT an open empirical question rather than a predictable transfer. Second, while test-time adaptation by entropy minimization is itself well established, prior work adapts model parameters; the alternative of adapting the *target speaker embedding* at test time, an option that exists only for speaker-conditioned tasks such as PVAD, has not to our knowledge been examined. Our study investigates both gaps, adapting the internal feature transformation via TTT (Section 3.3) and the speaker embedding via TEA (Section 3.4).

3. Proposed Method

We study two test-time adaptation mechanisms layered on top of the FDE-Mamba backbone. Section 3.1 briefly recaps the backbone. Section 3.3 introduces the VAD-gated TTT adapter, an in-model mechanism that adapts an internal weight matrix on the speaker-conditioned feature stream of each utterance. Section 3.4 introduces TEA, an inference-time procedure that adapts the target speaker embedding while keeping all model parameters frozen. Section 3.5 clarifies that the two mechanisms act at distinct representation levels, and Section 3.6 describes the post hoc Gaussian smoothing step that we treat as a separate, clearly delineated component.

Notation. We use bold lowercase for vectors and bold uppercase for matrices. A test utterance is a sequence of T frames. The acoustic input is $\mathbf{x}_{1:T}$ with per-frame feature $\mathbf{x}_t \in \mathbb{R}^F$, where F is the input feature dimension; $\mathbf{e} \in \mathbb{R}^E$ is the E -dimensional target speaker embedding (a d-vector). The backbone hidden dimension is D , so the latent and speaker-conditioned features $\mathbf{h}_t, \mathbf{F}_t, \tilde{\mathbf{F}}_t$ all lie in $\mathbb{R}^D$, and the TTT weight matrix $\mathbf{W} \in \mathbb{R}^{D \times D}$. Frame-level outputs are the VAD logits $\mathbf{v}_t \in \mathbb{R}^2$ (non-speech vs. speech) and the PVAD logits $\mathbf{p}_t \in \mathbb{R}^3$ (ns, ntss, tss). The subscript $[s]$ selects the speech component of a softmax output, $\odot$ denotes the element-wise (Hadamard) product, $(\cdot)^\top$ the matrix transpose, and $\|\cdot\|$ the Euclidean norm. A parenthesized superscript $(\cdot)^{(k)}$ indexes an inner-loop iteration. Table 1 summarizes the symbols.

Table 1. Summary of notation used in Section 3.

Symbol	Meaning
T	Number of frames in the utterance
F, E, D	Input-feature, speaker-embedding, and hidden dimensions
$\mathbf{x}_t \in \mathbb{R}^F$	Acoustic feature at frame t (log-Mel)
$\mathbf{e} \in \mathbb{R}^E$	Target speaker embedding (d-vector)
$\mathbf{h}_t \in \mathbb{R}^D$	Encoder latent representation
$\mathbf{F}_t \in \mathbb{R}^D$	Weighted-residual fused feature (Equation (1))
$\tilde{\mathbf{F}}_t \in \mathbb{R}^D$	Speaker-conditioned feature after FiLM (Equation (2))
$\gamma(\mathbf{e}), \beta(\mathbf{e}) \in \mathbb{R}^D$	FiLM scale and shift vectors
$\mathbf{v}_t, \mathbf{p}_t$	VAD logits ($\in \mathbb{R}^2$), PVAD logits ($\in \mathbb{R}^3$)
$p_{\text{vad}}(t), g_t$	VAD speech probability; VAD gate $g_t = p_{\text{vad}}(t)$
$q_{t,c}$	Softmax PVAD probability of class $c \in \{\text{ns}, \text{ntss}, \text{tss}\}$ at frame t
$\mathbf{W}, \mathbf{W}_0 \in \mathbb{R}^{D \times D}$	TTT weight matrix; its learned initialization
$\mathbf{z}_t \in \mathbb{R}^D$	TTT adapter output at frame t
η, α, b	TTT inner-loop rate (learned), EMA coefficient ($\alpha = 0.7$), chunk size ($b = 16$)
μ, K, λ	TEA learning rate, step count, joint-loss weight ($\lambda = 0.5$)
σ	Gaussian-smoothing standard deviation

Figure 2 contrasts the two mechanisms at a conceptual level before we detail them. They differ in what they adapt and when: TTT updates a weight matrix *inside* the model as it processes each sequence, whereas TEA keeps the model frozen and updates the speaker embedding *fed into* it, once per utterance.

3.1. FDE-Mamba Backbone

The FDE-Mamba backbone [9], whose flowchart is depicted in Figure 3, follows the two-stage design of FDE-RNN [8], comprising a Dynamic Encoder for VAD and a detachable Personalization module for PVAD, with all three recurrent components instantiated as Mamba [19] blocks. Given a sequence of acoustic features $\mathbf{x}_{1:T}$ with $\mathbf{x}_t \in \mathbb{R}^F$ and a target speaker embedding $\mathbf{e} \in \mathbb{R}^E$, the backbone produces frame-level VAD logits $\mathbf{v}_t$ and PVAD logits $\mathbf{p}_t$.

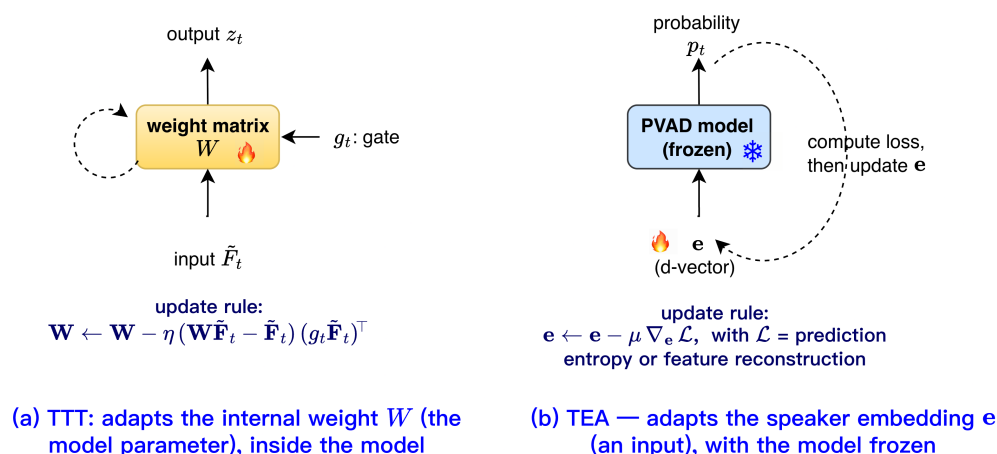


Figure 2. Conceptual contrast of the two test-time adaptation mechanisms. (a) TTT treats a weight matrix W as a fast state that is updated by a self-supervised step (gated by the VAD probability g_t) while the model processes the feature stream $\tilde{F}_t$; the model parameter changes during inference. (b) TEA freezes all model parameters and instead updates the target speaker embedding e once per utterance by minimizing a self-supervised loss $\mathcal{L}$ on the predictions. The two thus adapt different quantities (a parameter vs. an input) at different granularities.

Following Figure 3, the per-frame data flow inside the dynamic encoder is as follows. The Prediction Mamba receives $\mathbf{x}_t \oplus h_{t-1}$ as its per-frame input and outputs the VAD speech probability $p_{\text{vad}}(t) = \text{softmax}(\mathbf{v}_t)_{[s]}$, the speech-class entry of the softmax over the VAD logits; its horizontal arrows $s_{t-1} \rightarrow s_t$ carry the block's internal SSM state across frames, analogous to $h_{t-1} \rightarrow h_t$ for the Encoder Mamba. The same scalar $p_{\text{vad}}(t)$ is then routed to three destinations: it forms the two-way logits $(p_{\text{vad}}, -p_{\text{vad}})$ used to compute $\mathcal{L}_{\text{vad}}$; it drives the Control block, which either connects h_{t-1} to the Encoder Mamba on speech frames or bypasses the encoder via the dashed "copy" path on non-speech frames; and it weights the residual connection that combines the encoder output with the raw input:

$$\mathbf{F}_t = \mathbf{h}_t + (1 - p_{\text{vad}}(t)) \mathbf{x}_t. \quad (1)$$

The fused feature is then modulated by the speaker embedding through a Feature-wise Linear Modulation (FiLM) layer [5]:

$$\tilde{\mathbf{F}}_t = \gamma(\mathbf{e}) \odot \mathbf{F}_t + \beta(\mathbf{e}), \quad (2)$$

where $\gamma(\mathbf{e})$ and $\beta(\mathbf{e})$ are the scale and shift vectors generated from $\mathbf{e}$, and $\odot$ denotes the element-wise product. The speaker-conditioned feature $\tilde{\mathbf{F}}_t$ is processed by the back-end Mamba block and a linear classifier to produce the PVAD logits $\mathbf{p}_t \in \mathbb{R}^3$, shown at the top of Figure 3. We emphasize that $\mathbf{p}_t$ (the 3-dimensional PVAD output over {ns, ntss, tss}) and $p_{\text{vad}}(t)$ (the scalar VAD speech probability produced by the Prediction Mamba) are distinct quantities that appear at different points in the figure. In the original FDE-Mamba, $\tilde{\mathbf{F}}_{1:T}$ passes directly into the back-end block; in this work, we insert the VAD-Gated TTT adapter at exactly this point, so that the adapter operates on the speaker-conditioned stream.

3.2. Overview of the Proposed Modifications

We modify the personalization module of FDE-Mamba at two points, as illustrated in Figure 4. First, within the interaction block, we insert a *VAD-gated TTT adapter* between the FiLM layer and the Mamba block. This adapter maintains an internal weight matrix W that is updated at every chunk by a self-supervised reconstruction signal, gated by the VAD probability g_t so that only speech frames drive the adaptation. It adapts the

speaker-conditioned feature stream $\tilde{\mathbf{F}}_{1:T}$ to the acoustic and speaker conditions of the current test utterance. Second, upstream of the interaction block, we introduce TEA (Test-time Embedding Adaptation), which adapts the target speaker embedding $\mathbf{e}_{\text{target}}$ itself before it enters the FiLM generator. With all model parameters frozen, TEA performs a small number of gradient steps on $\mathbf{e}$ using self-supervised objectives computed from the model's output, directly targeting the enrollment–test mismatch.

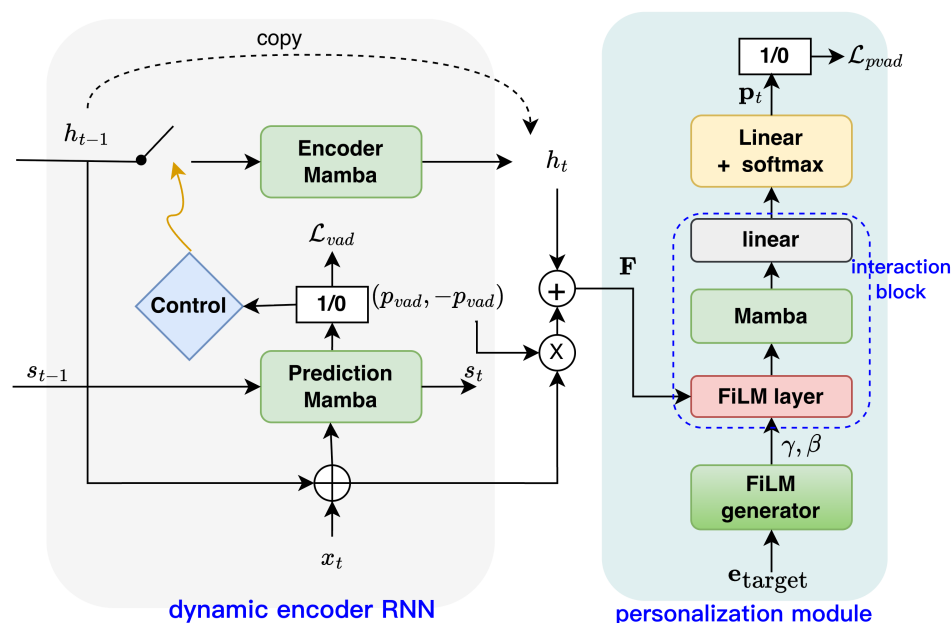


Figure 3. The original FDE-Mamba flowchart [9], comprising a Dynamic Encoder RNN (left) for VAD and a personalization module (right) for PVAD. Within the dynamic encoder, the Prediction Mamba receives $\mathbf{x}_t \oplus \mathbf{h}_{t-1}$ as its per-frame input and produces the VAD speech probability $p_{\text{vad}}(t)$; the horizontal arrows $s_{t-1} \rightarrow s_t$ denote the Mamba block's internal SSM state carried across frames, analogous to $h_{t-1} \rightarrow h_t$ for the Encoder Mamba. The scalar $p_{\text{vad}}(t)$ is used at three places: (i) it forms the two-way logits $(p_{\text{vad}}, -p_{\text{vad}})$ that supply $\mathcal{L}_{\text{vad}}$; (ii) it drives the Control block, whose 1/0 decision either connects h_{t-1} to the Encoder Mamba (speech) or bypasses the encoder via the dashed “copy” path (non-speech); and (iii) it weights the residual $(1 - p_{\text{vad}}(t)) \mathbf{x}_t$ added to the encoder output $\mathbf{h}_t$ to form the fused feature $\mathbf{F}$ (Equation (1)). The symbol shown at the top of the personalization module is $\mathbf{p}_t \in \mathbb{R}^3$, the frame-level PVAD output over the classes $\{\text{ns}, \text{ntss}, \text{tss}\}$; it is a distinct quantity from the scalar $p_{\text{vad}}(t)$ produced by the Prediction Mamba. The interaction block (blue dashed box) is the locus in which we insert the VAD-gated TTT adapter; see Figure 4 for the updated personalization module.

The two modifications thus act at different points in the data flow: the TTT adapter operates *inside* the model on the feature representation, while TEA operates on the *input* embedding. Their conceptual contrast is shown in Figure 2; Figure 4 shows their concrete placement within the personalization module. Additionally, a Gaussian smoothing step (Section 3.6) is applied as model-agnostic post-processing on the output probabilities; it is unrelated to the adaptation mechanisms and is reported separately. Sections 3.3 and 3.4 detail each adaptation mechanism in turn.

3.3. VAD-Gated Test-Time Training Adapter

Background: TTT-Linear. The TTT-Linear layer [13] treats its hidden state as a linear model whose weight matrix $\mathbf{W}$ is updated by a step of self-supervised learning at each position. For an input token $\mathbf{u}_t \in \mathbb{R}^D$, the weight is updated by gradient descent on a

reconstruction loss $\ell(\mathbf{W}; \mathbf{u}_t) = \|\mathbf{W}\mathbf{u}_t - \mathbf{u}_t\|^2$, which measures how well the linear map $\mathbf{W}$ reconstructs its own input:

$$\mathbf{W}_t = \mathbf{W}_{t-1} - \eta \nabla_{\mathbf{W}} \ell(\mathbf{W}_{t-1}; \mathbf{u}_t), \quad \mathbf{z}_t = \mathbf{W}_t \mathbf{u}_t, \quad (3)$$

where η is the inner-loop learning rate. Because $\mathbf{W}$ continues to be updated on the test sequence, the layer adapts its representation to the local statistics of each input even at inference time. Directly inserting Equation (3) into the PVAD personalization pathway, however, raises three design questions specific to this task. We address each in turn.

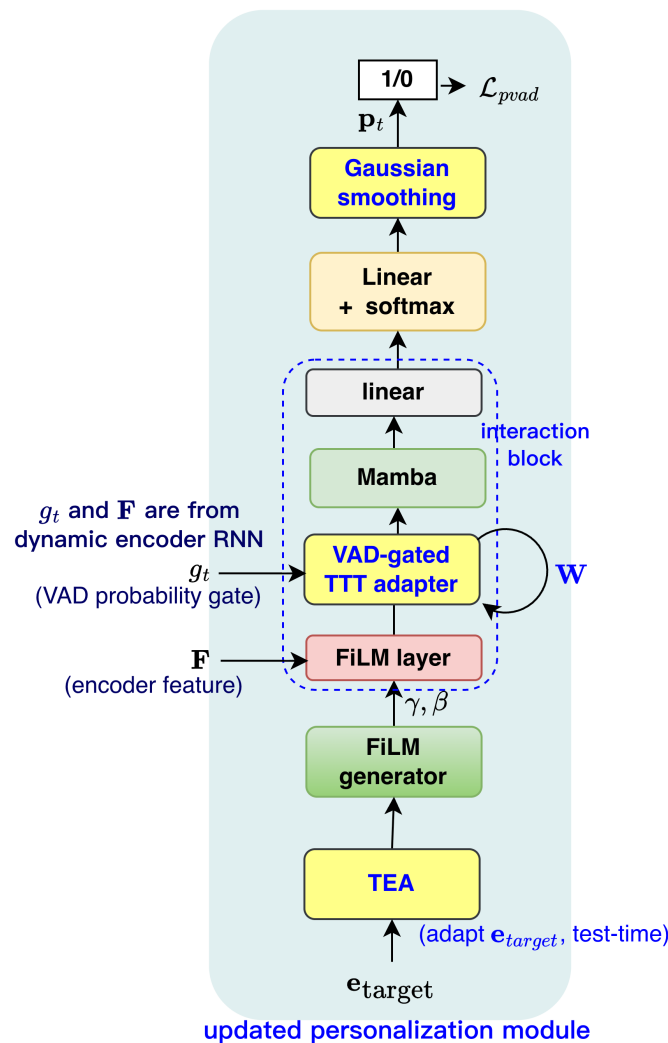


Figure 4. The updated personalization module of FDE-Mamba, incorporating the two proposed test-time adaptation mechanisms (highlighted in blue). The target speaker embedding $\mathbf{e}_{\text{target}}$ enters TEA, which adapts the embedding at inference time before passing it to the FiLM generator; the generator produces the modulation parameters γ, β . The encoder feature $\mathbf{F}$ and γ, β are then fused by the FiLM layer. Within the interaction block (blue dashed box), we insert the VAD-gated TTT adapter between the FiLM layer and the Mamba block: it adapts an internal weight matrix $\mathbf{W}$ (via the EMA self-update loop) on the speaker-conditioned feature stream, gated by the VAD probability g_t supplied by the dynamic encoder outside this module. The Gaussian smoothing post-processing step is added after the Linear + softmax layer. Components introduced in this work are shown in bold blue.

3.3.1. Restricting Adaptation to Speech Frames via a VAD Gate

In a generic sequence model, every position contributes equally to the inner-loop update. In PVAD, however, a large fraction of frames are non-speech, and the reconstruction signal on those frames reflects background noise rather than speaker-relevant structure. Updating $\mathbf{W}$ uniformly across all frames would therefore let non-speech frames steer the adaptation, diluting the speaker-conditioned information the adapter is meant to refine.

We address this by gating the inner-loop update with the VAD probability. Operating on the speaker-conditioned stream $\tilde{\mathbf{F}}_{1:T}$ (Equation (2)), and writing $g_t = p_{\text{vad}}(t)$ for the (detached) VAD probability at frame t —the speech-class softmax output of the prediction block, the same quantity used in the weighted residual of Equation (1)—the gated update direction is

$$\Delta \mathbf{W}_t = \frac{1}{\sqrt{D}} (\mathbf{W} \tilde{\mathbf{F}}_t - \tilde{\mathbf{F}}_t) (g_t \tilde{\mathbf{F}}_t)^\top, \quad (4)$$

where D is the hidden dimension and the factor $1/\sqrt{D}$ scales the update to keep its magnitude stable across feature dimensions, so that the contribution of frame t to the update is scaled by its probability of being speech. Frames the VAD stage deems non-speech ($g_t \approx 0$) contribute little to $\Delta \mathbf{W}_t$, while confident speech frames drive the adaptation. We refer to this as a *VAD-gated* adapter, since the VAD probability g_t acts as a gate that confines the test-time weight update to frames the model itself judges to be speech. This couples the adapter to the model's own VAD decision rather than to an external mask, and keeps the adaptation focused on the frames where speaker discrimination matters.

To keep the inner-loop update stable when adapting over long utterances, we additionally clamp the update with a scaled hyperbolic tangent applied element-wise to $\Delta \mathbf{W}_t$,

$$\Delta \mathbf{W}_t \leftarrow c \tanh(\Delta \mathbf{W}_t / c), \quad c = 0.1, \quad (5)$$

which bounds every entry of the update to $(-c, c)$ and limits the influence of outlier frames whose reconstruction error is large. We found this bounded update to be important for stability: without it, $\mathbf{W}$ can accumulate large updates and drift over a long utterance.

3.3.2. Stabilizing the Weight Trajectory with an EMA Update

In TTT-Linear the weight matrix is overwritten at each step (Equation (3)). Applied unmodified during inference on PVAD, this can let $\mathbf{W}$ drift over a long utterance, since each update is taken on a single, noisy reconstruction signal and there is no outer-loop loss to correct accumulated error. To stabilize the trajectory, we combine the gradient step with an exponential moving average (EMA). The post-step weight applies the (clamped) update of Equation (5) with the inner-loop rate (the factor of 2 arises from differentiating the squared reconstruction loss $\ell = \|\mathbf{W} \tilde{\mathbf{F}}_t - \tilde{\mathbf{F}}_t\|^2$):

$$\mathbf{W}^{\text{step}} = \mathbf{W} - 2\eta \Delta \mathbf{W}_t. \quad (6)$$

The retained weight is then a convex combination of the previous and updated matrices,

$$\mathbf{W} \leftarrow \alpha \mathbf{W} + (1 - \alpha) \mathbf{W}^{\text{step}}, \quad (7)$$

where $\alpha \in [0, 1)$ is the EMA coefficient. A larger α retains more of the accumulated state and yields a smoother trajectory; $\alpha = 0$ recovers the plain TTT step. The adapter output applied to the feature stream uses the post-step weight,

$$\mathbf{z}_t = \tilde{\mathbf{F}}_t + \mathbf{W}^{\text{step}} \tilde{\mathbf{F}}_t, \quad (8)$$

i.e., the adapter produces an additive correction to the speaker-conditioned feature rather than replacing it, so that the back-end block always receives the original feature plus a learned, input-dependent residual. The EMA-smoothed weight $\mathbf{W}$ of Equation (7) is carried to the next chunk as the running adaptation state, while the current chunk's output uses the post-step weight $\mathbf{W}^{\text{step}}$.

3.3.3. Chunked Updates for Streaming Operation

Performing one gradient step per frame (Equation (3)) is both expensive and unstable on the relatively short, noisy feature stream of a PVAD utterance. We therefore divide the sequence into non-overlapping chunks of length b and perform a single matrix-form update per chunk, accumulating the gated reconstruction signal over the b frames within the chunk before applying Equations (6) and (7). The resulting weight $\mathbf{W}$ is carried across chunks within an utterance, providing a running adaptation state, and is reset to its learned initialization $\mathbf{W}_0$ at the start of each utterance. The chunk size b trades off adaptation granularity against update stability, and is treated as a hyperparameter in our study (Section 5).

Taken together, Equations (4)–(8) define the VAD-gated TTT adapter. We emphasize that the adapter operates on $\tilde{\mathbf{F}}_{1:T}$, the feature stream after speaker fusion. The distribution shift it absorbs at test time is therefore a composite of acoustic conditions, speaking style, and speaker characteristics as they are jointly expressed in the speaker-conditioned representation; the adapter does not, and is not intended to, isolate any single one of these factors.

3.4. TEA: Test-Time Embedding Adaptation

The second mechanism, Test-time Embedding Adaptation (TEA), adapts a different part of the system. Rather than modifying any model parameter, it adapts the target speaker embedding $\mathbf{e}$ itself, directly addressing the case where the enrollment embedding is mismatched with the test signal. With all model parameters frozen, the embedding is treated as the only free variable. Starting from the enrolled embedding $\mathbf{e}^{(0)} = \mathbf{e}$, TEA performs K gradient steps on a self-supervised objective $\mathcal{L}_{\text{TEA}}$ computed from the test utterance,

$$\mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mu \nabla_{\mathbf{e}} \mathcal{L}_{\text{TEA}}(\mathbf{e}^{(k)}), \quad k = 0, \dots, K-1, \quad (9)$$

and then runs a final forward pass with the adapted embedding $\mathbf{e}^{(K)}$ to produce the predictions. Because the procedure adapts only the input embedding and resets it for each utterance, it introduces no persistent change to the model and no additional parameters. In Equation (9), $\mathcal{L}_{\text{TEA}}$ is a self-supervised objective requiring no ground-truth labels; we consider three instantiations:

Entropy. Following the entropy-minimization principle [10], we minimize the uncertainty of the PVAD predictions, restricted to speech frames via the VAD gate g_t :

$$\mathcal{L}_{\text{ent}} = \frac{\sum_t g_t [-\sum_c q_{t,c} \log q_{t,c}]}{\sum_t g_t}, \quad (10)$$

where $q_{t,c} = \text{softmax}(\mathbf{p}_t)_c$ is the predicted PVAD probability for class c at frame t . Minimizing $\mathcal{L}_{\text{ent}}$ sharpens the model's decisions on speech frames, encouraging an embedding under which the model is confident.

Reconstruction. The second objective reuses the self-supervised signal of the TTT adapter. With the speaker-conditioned feature $\tilde{\mathbf{F}}_t$ recomputed from the candidate embed-

ding via Equations (1) and (2), the loss is the reconstruction error of the adapter's initial linear map,

$$\mathcal{L}_{\text{rec}} = \frac{1}{T} \sum_t \|\mathbf{W}_0 \tilde{\mathbf{F}}_t - \tilde{\mathbf{F}}_t\|^2, \quad (11)$$

which favors an embedding whose induced feature stream is consistent with the representation the adapter was trained to reconstruct.

Joint. The third objective is a weighted combination,

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{ent}} + \lambda \mathcal{L}_{\text{rec}}, \quad (12)$$

with λ a balancing coefficient, set to $\lambda = 0.5$ in our experiments. We compare all three objectives empirically in Section 5.

Although TEA shares the entropy-minimization principle with prior test-time adaptation methods [10], it differs in what is adapted: where those methods update model parameters (typically normalization or affine weights), TEA updates the target speaker embedding. This choice is specific to speaker-conditioned tasks and, to our knowledge, has not previously been studied for PVAD.

3.5. Two Complementary Adaptation Levels

The two mechanisms adapt distinct parts of the system and should not be read as redundant. The VAD-gated TTT adapter modifies an internal weight matrix that transforms the speaker-conditioned feature stream; its effect is a learned, input-dependent residual applied to $\tilde{\mathbf{F}}_{1:T}$ within the model. TEA, by contrast, leaves every model weight untouched and moves the speaker embedding $\mathbf{e}$, which enters the model upstream through the FiLM layer (Equation (2)). One mechanism therefore operates in the feature-transformation space and the other in the speaker-representation space. We avoid characterizing them as cleanly handling separate sources of distribution shift—for instance attributing noise to one and speaker mismatch to the other—because the speaker-conditioned representation entangles these factors; rather, we describe them as acting at different representation levels and evaluate their independent and combined effects directly.

3.6. Gaussian Smoothing Post-Processing

Finally, we consider a post hoc smoothing step that is independent of the two adaptation mechanisms and of the model. After the per-frame probabilities are produced, a one-dimensional Gaussian filter of standard deviation σ is applied along the time axis to both the VAD and PVAD probability sequences prior to thresholding:

$$\hat{q}_{t,c} = \sum_{\tau} \mathcal{K}_{\sigma}(\tau) q_{t-\tau,c}, \quad \mathcal{K}_{\sigma}(\tau) \propto \exp\left(-\frac{\tau^2}{2\sigma^2}\right). \quad (13)$$

This suppresses isolated single-frame errors—brief false detections within non-speech regions and momentary dropouts within speech regions—by enforcing temporal continuity in the probability sequence. We stress that Gaussian smoothing is a generic post-processing operation unrelated to test-time adaptation; we include it because it appears in our evaluation pipeline, and we report its contribution separately (Section 5) so that it is not conflated with the effect of the TTT adapter or TEA.

4. Experimental Setup

4.1. Dataset and Data Preparation

We follow the experimental protocol of FDE-RNN [8] and FDE-Mamba [9] exactly, to ensure direct comparability with prior work. All models are trained and evaluated on the

LibriSpeech corpus [23]. To simulate multi-speaker scenarios with speaker turns, we adopt the concatenated-utterance scheme of Ding et al. [3]: for each sample, the number of speakers is drawn from [1, 3], non-repeating speakers are randomly selected with one designated as the target, a single utterance is sampled per speaker, and the utterances are shuffled and concatenated. Target speaker embeddings are 256-dimensional d-vectors [15] obtained by passing additional, independently sampled utterances of the target speaker through a pre-trained speaker verification model. Following Yu et al. [16], in 20% of training samples the designated target is replaced by a speaker absent from the concatenated utterance, encouraging robustness to unseen enrollment conditions. Multi-condition training with reverberation is applied, with room impulse responses and additive noise drawn from standard corpora [24]. There is no speaker or utterance overlap between the training and test partitions, and all embeddings are constructed within their respective partitions.

4.2. Evaluation Metrics

Consistent with prior work [3,8], we report: **Average Precision (AP)** for each category, presented as the combined *ns&ntss* AP and the *tss* AP; **mean Average Precision (mAP)**, the average of AP across categories, used as the primary ranking metric; **Accuracy**, the frame-level classification accuracy; and **Recall** and **Precision** for the *tss* category, to characterize target-speaker detection sensitivity and specificity. To assess inference cost, we additionally report the **real-time factor (RTF)**, peak memory allocation, and kilo-FLOPs per frame.

4.3. Implementation Details

All models are implemented in PyTorch (version 2.11.0). Input features are 40-dimensional log-Mel filterbank coefficients, and the hidden dimension of all blocks is set to $D = 64$, following [9]. We train with a batch size of 64 and an outer-loop learning rate of 10^{-3} . The VAD-gated TTT adapter uses a chunk size of $b = 16$ and an EMA coefficient of $\alpha = 0.7$; the inner-loop learning rate η is a learned scalar (a trainable parameter initialized to 0, updated by backpropagation during outer-loop training). For TEA, the embedding learning rate μ , number of steps K , and loss type (entropy, reconstruction, or joint) are swept as described in Section 5; unless otherwise stated, the Gaussian smoothing standard deviation is $\sigma = 5$, applied to both the VAD and PVAD probability sequences. The same adaptation mechanisms are additionally applied to the LSTM-based FDE-RNN backbone to assess cross-architecture generality. Unless otherwise stated, all systems are evaluated with the Gaussian smoothing post-processing described in Section 3.6. We treat this smoothing as a standard, model-agnostic component of the inference pipeline, and focus our analysis on the behavior of the TTT and TEA adaptation mechanisms.

5. Results and Discussion

We first establish the overall effect of the proposed test-time adaptation against the FDE-Mamba backbone (Section 5.1). We then dissect the contribution of each component through an ablation study (Section 5.2), report the computational cost (Section 5.3), verify that the same mechanisms transfer to the LSTM-based FDE-RNN backbone (Section 5.4), and compare against TENT as an existing test-time adaptation baseline (Section 5.5). We close with a discussion of a recall–precision trade-off and the limitations of the study (Section 5.6), and, in a controlled distribution-shift analysis (Section 5.7), examine how the observed gains scale with the amount of mismatch actually present at inference time. Throughout, “TTT” denotes the VAD-gated TTT adapter, “TEA” the test-time embedding adaptation, and “Gauss” Gaussian smoothing with $\sigma = 5$.

5.1. Main Results

Our primary comparison is against the FDE-Mamba backbone [9], which is the direct, already-published baseline that this work extends. Table 2 reports the FDE-RNN baseline for reference, the FDE-Mamba backbone, and the proposed full system (TTT + TEA + Gauss). The proposed system improves the primary PVAD metric, mAP, from 0.9605 to 0.9641 (+0.0036) over FDE-Mamba, with the ns&ntss AP rising from 0.9699 to 0.9740 and precision from 0.881 to 0.899; accuracy improves from 89.87% to 90.19%. We state plainly that the improvement over the FDE-Mamba backbone is modest in magnitude. Its value, in the context of this empirical study, lies in showing that inference-time adaptation—which requires no retraining and no additional labels—yields a consistent gain on a strong, recently published PVAD baseline, and in characterizing precisely where that gain comes from, which we do next. To check that the best-mAP result is not an artifact of the stochasticity in the TEA inner loop, we repeated the evaluation of the proposed system five times with the trained model held fixed; the PVAD mAP was 0.96406 ± 0.00001 (mean $\pm$ std over five runs), confirming that the test-time adaptation procedure is highly stable at inference. We emphasize that this quantifies only the inference-time stochasticity, not the variance across independent training runs (see Section 5.6). The clean-test gain reported here is small in absolute terms; in Section 5.7 we show that this gain grows systematically with the amount of distribution shift present at inference, reaching +0.033 in mAP under strong enrollment mismatch, and that a coherent precision-oriented adaptation signature emerges across four metrics.

Table 2. Main results on the LibriSpeech PVAD task. The proposed system applies the VAD-gated TTT adapter, TEA, and Gaussian smoothing on top of the FDE-Mamba backbone; the ablation in Table 3 decomposes the individual contributions of TTT, TEA, and smoothing. Improvements (Δ) are reported relative to the FDE-Mamba backbone. Best value in each column is in **bold**.

Model	Acc. (%)	Recall	Prec.	ns & ntss	tss	mAP
FDE-RNN [8] (reproduced)	86.85	0.847	0.834	0.9587	0.9101	0.9419
FDE-Mamba [9] (baseline)	89.87	0.870	0.881	0.9699	0.9452	0.9605
Proposed (TTT + TEA + Gauss)	90.19	0.856	0.899	0.9740	0.9477	0.9641
Δ vs. FDE-Mamba	+0.32	−0.014	+0.018	+0.0041	+0.0025	+0.0036

Table 3. Ablation of the three components on the FDE-Mamba backbone. ✓ indicates the component is included in that configuration. The baseline (top row, no adaptation) has 145,524 parameters; all other variants, which include the TTT adapter, have 150,053 parameters. Best value in each column is in **bold**.

TTT	TEA	Gauss	Acc. (%)	Recall	Prec.	ns & ntss	tss	mAP
			89.87	0.870	0.881	0.9699	0.9452	0.9605
		✓	90.47	0.864	0.899	0.9694	0.9469	0.9618
✓			89.99	0.849	0.900	0.9723	0.9460	0.9623
✓		✓	90.08	0.853	0.900	0.9734	0.9464	0.9632
	✓		89.31	0.835	0.890	0.9672	0.9388	0.9564
	✓	✓	89.41	0.839	0.892	0.9686	0.9413	0.9578
✓	✓		90.15	0.851	0.902	0.9732	0.9460	0.9633
✓	✓	✓	90.19	0.856	0.899	0.9740	0.9477	0.9641

5.2. Ablation Study

To attribute the overall gain to its sources, Table 3 reports every combination of the three components on the FDE-Mamba backbone. We make four observations.

The TTT adapter contributes a small, precision-oriented gain. Adding TTT alone raises mAP from 0.9605 to 0.9623 (+0.0018) and precision from 0.881 to 0.900, while accuracy is essentially unchanged. The adapter sharpens speaker discrimination rather than improving overall frame classification.

TEA helps only in combination with TTT. Used on its own (“+TEA”), TEA does not improve over the backbone—mAP decreases to 0.9564—indicating that the inference-time perturbation of the embedding is not, by itself, beneficial under the current protocol. Added on top of TTT, however, it raises mAP from 0.9623 to 0.9633, and the full combination attains the best mAP of 0.9641. We report this dependence explicitly: the value of TEA in this study is conditional on the presence of the TTT adapter.

Gaussian smoothing is the largest single contributor to accuracy, and is independent of the adaptation mechanisms. Smoothing alone raises accuracy from 89.87% to 90.47%—the largest accuracy improvement of any single component, and larger than that of the TTT adapter—while its effect on mAP is comparatively small (+0.0013). Because smoothing is a generic, model-agnostic post-processing step, we report its contribution separately so that it is not conflated with the adaptation mechanisms: the accuracy gains it provides and the precision/mAP gains the adapter provides arise from different mechanisms.

A recall–precision trade-off accompanies adaptation. Across the table, adding TTT raises precision but lowers tss recall (0.870 → 0.849); TEA and smoothing partially restore it (to 0.856), but the full system does not recover the recall of the unadapted backbone. We return to this in Section 5.6.

A sweep over the TEA objective (entropy, reconstruction, joint), learning rate μ , and step count K is reported in Tables 4 and 5; we select the configuration with the highest mAP for the main results. Two findings stand out. First, TEA is highly sensitive to the learning rate: all three objectives degrade sharply as μ increases, collapsing by $\mu = 10^{-1}$ (the entropy objective falls to 0.526). A small learning rate ($\mu = 10^{-3}$) is essential, with the reconstruction and joint objectives performing best there (mAP 0.9641). Second, more adaptation steps do not help: mAP peaks at $K = 2$ and declines thereafter, dropping to 0.9567 at $K = 10$ (Table 5). Both findings indicate that TEA is useful only as a small, carefully bounded correction; over-adaptation harms performance.

Table 4. TEA objective and learning-rate sweep on FDE-Mamba (PVAD mAP, applied on top of the TTT adapter, $K = 2$ steps). Best mAP value in **bold**.

Objective	$\mu = 10^{-3}$	5×10^{-3}	10^{-2}	5×10^{-2}	10^{-1}
Entropy	0.9610	0.9387	0.9194	0.7062	0.5257
Reconstruction	0.9641	0.9551	0.8245	0.4484	0.4578
Joint	0.9641	0.9563	0.8424	0.4485	0.4581

Table 5. Effect of the TEA step count K on FDE-Mamba (PVAD mAP, reconstruction objective, $\mu = 10^{-3}$). Best value in **bold**.

Steps K	1	2	5	10
PVAD mAP	0.9639	0.9641	0.9633	0.9567

5.3. Computational Cost

Table 6 reports the inference cost. The TTT adapter is inexpensive: it adds about 3% more parameters (145,524 → 150,053) and raises the real-time factor (RTF) only modestly (0.00519 → 0.00708, a $1.4\times$ increase), with no change in FLOPs per frame. TEA, however, is costly. Because it performs K gradient steps on the embedding—each requiring a forward and backward pass through the model—per utterance at inference, both RTF and FLOPs

scale with K . At the chosen setting $K = 2$, RTF rises to 0.02876 ($5.5\times$ the FDE-Mamba backbone) and FLOPs per frame increase roughly sevenfold ($221.9 \rightarrow 1553.5$). We note plainly that this nearly erases the RTF advantage of the Mamba backbone over FDE-RNN (0.02813): the full system with TEA runs at approximately the same speed as the LSTM baseline. The peak memory also grows substantially ($43.5 \text{ MB} \rightarrow 136 \text{ MB}$), reflecting the activations retained for the backward pass. This cost–benefit profile is unfavorable: TEA at $K = 2$ buys a further $+0.0008$ mAP over TTT alone while increasing RTF by roughly $4\times$. The TTT adapter, by contrast, delivers most of the benefit at negligible cost. We return to this trade-off in Section 5.6.

Table 6. Inference cost (RTF measured on GPU with warm cache). The TTT adapter is nearly free; TEA cost scales with the step count K .

Model	Params	RTF	Peak Mem. (MB)	kFLOPs/Frame
FDE-RNN (reproduced)	92,372	0.02813	16.60	136.43
FDE-Mamba	145,524	0.00519	43.45	221.93
+TTT	150,053	0.00708	54.72	221.93
+TTT + TEA ($K = 1$)	150,053	0.01845	135.97	887.73
+TTT + TEA ($K = 2$)	150,053	0.02876	135.97	1553.52
+TTT + TEA ($K = 5$)	150,053	0.06137	134.97	3550.90

5.4. Generalization to the FDE-RNN Backbone

While the focus of this study is the FDE-Mamba backbone, we additionally verify that the proposed mechanisms are not specific to a selective state-space model by applying the same TTT adapter and TEA to the LSTM-based FDE-RNN. Table 7 shows the same qualitative pattern: TTT raises mAP from 0.9419 to 0.9434, and the full combination reaches 0.9443, with tss AP trending upward overall as components are added. The gains are again modest and of comparable magnitude to those on FDE-Mamba. This consistency across an LSTM and a selective state-space backbone indicates that the contribution of the adaptation mechanisms is largely independent of the underlying sequence model. We present this as a supporting observation rather than a primary result.

Table 7. Generalization check: the same adaptation mechanisms applied to the FDE-RNN backbone. The reconstruction objective is used for TEA. Best value in each column is in **bold**.

Model	Acc. (%)	Recall	Prec.	ns & ntss	tss	mAP
FDE-RNN (reproduced)	86.85	0.847	0.834	0.9587	0.9101	0.9419
+TTT	87.17	0.853	0.836	0.9583	0.9136	0.9434
+TTT + Gauss	87.53	0.858	0.841	0.9598	0.9151	0.9439
+TTT + TEA	87.25	0.859	0.834	0.9585	0.9148	0.9440
+TTT + TEA + Gauss	87.55	0.863	0.842	0.9602	0.9168	0.9443

5.5. Comparison with an Existing TTA Baseline

To situate the proposed mechanisms within the broader test-time adaptation literature, we additionally compare against TENT [10], the canonical entropy-minimization TTA method. TENT freezes the trained model at inference time and updates only the affine parameters of the model’s normalization layers to minimize the entropy of the predictions. It therefore adapts a different quantity than either of our mechanisms: where TTT adapts an internal weight matrix on the speaker-conditioned feature stream (Section 3.3) and TEA adapts the target speaker embedding (Section 3.4), TENT adapts a set of task-agnostic normalization affines. Comparing against TENT therefore addresses a well-defined method-

ological question: whether the PVAD-specific adaptation surfaces we target confer any advantage over the standard normalization-affine adaptation of the TTA literature.

Implementation. We follow the original TENT protocol. All parameters of the FDE-Mamba backbone are frozen except the weight and bias of every LayerNorm layer in the model—namely the three pre-norms of the Encoder Mamba, Prediction Mamba, and interaction-block Mamba, for a total of 384 scalar parameters. For each test batch we take $K = 1$ Adam step at learning rate 10^{-3} on the entropy of the PVAD softmax; the affine parameters are then reset to their trained values before the next batch, matching the per-batch reset used by TEA elsewhere in this paper. For consistency with the VAD-gating convention used by TTT and TEA, the entropy is weighted by the detached VAD speech probability. All configurations reported below include the same Gaussian smoothing ($\sigma = 5$) as the other rows of Sections 5.1 and 5.2.

Results. Table 8 reports two comparisons on the FDE-Mamba backbone: TENT applied on top of the baseline (top block), and TENT applied on top of our full proposed system (bottom block). This is a 2×2 design that answers two related questions—whether TENT alone helps the baseline, and whether TENT adds anything when stacked on top of the full proposed system—while sharing the same Gaussian smoothing ($\sigma = 5$) and the same VAD-gated entropy computation across all four cells.

Table 8. Comparison with TENT [10] on the FDE-Mamba backbone. All rows include Gaussian smoothing ($\sigma = 5$). Top block: TENT applied on top of the baseline. Bottom block: TENT applied on top of the proposed full system (TTT + TEA). Best value in each column is in **bold**.

Model	Acc. (%)	Recall	Prec.	ns & ntss	tss	mAP
FDE-Mamba + Gauss (baseline)	90.47	0.864	0.899	0.9694	0.9469	0.9618
+TENT	89.84	0.886	0.870	0.9667	0.9426	0.9524
Proposed (TTT + TEA + Gauss)	90.19	0.856	0.899	0.9740	0.9477	0.9641
+TENT	90.49	0.872	0.893	0.9711	0.9463	0.9619

Two observations from this table are as follows:

- First, applied on top of the FDE-Mamba baseline (top block), TENT does not improve PVAD mAP. mAP decreases from 0.9618 to 0.9524 (-0.0094), and precision decreases from 0.899 to 0.870 (-0.029), with a corresponding rise in recall from 0.864 to 0.886 ($+0.022$). Under the present evaluation protocol, and using the original TENT hyperparameters, updating LayerNorm affines to minimize prediction entropy does not act as a beneficial addition to the FDE-Mamba PVAD backbone.
- Second, applied on top of the full proposed system (bottom block), TENT does not provide an additive benefit either. mAP decreases from 0.9641 to 0.9619 (-0.0022), with a similar precision–recall shift (precision $0.899 \rightarrow 0.893$, recall $0.856 \rightarrow 0.872$). Both the ns&ntss AP and the tss AP decrease under the addition of TENT ($0.9740 \rightarrow 0.9711$ and $0.9477 \rightarrow 0.9463$), and the full proposed system without TENT remains the best-mAP configuration. Stacking TENT on top of our mechanisms is therefore neither redundant in the sense of leaving mAP unchanged, nor complementary in the sense of improving it further; the entropy signal that TENT relies on and the adaptation surfaces that TTT and TEA target do not combine additively in this backbone.

A note on scope. We use TENT’s default hyperparameters ($K = 1$, $\text{lr} = 10^{-3}$) without an extended sweep, so a better operating point for TENT may exist. We also do not compare against more recent entropy-minimization variants, which use adaptive sample filtering and different reset schedules and may behave differently. So what the comparison shows is more limited than a full TTA benchmark: TENT in its original form does not improve

on the FDE-Mamba PVAD baseline in our setup, and does not add anything on top of the proposed system. A broader TTA comparison for PVAD would be a natural next step.

5.6. Discussion and Limitations

The recall–precision trade-off. Adding the TTT adapter raises precision (0.881 $\rightarrow$ 0.900) but lowers tss recall (0.870 $\rightarrow$ 0.849): the adapter makes the model more conservative in declaring a frame to be target-speaker speech, committing fewer false positives but missing more true target frames. TEA and Gaussian smoothing partially recover recall (to 0.856 in the full system), but the full system still does not match the recall of the unadapted backbone. This trade-off is consistent with the precision-oriented nature of the reconstruction-based adaptation signal, and it is the main reason the accuracy gains are smaller than the AP gains. Applications that prioritize not missing target speech may prefer the unadapted backbone, while those that prioritize suppressing false target detections may benefit from the adapter.

Cost versus benefit of TEA. The two mechanisms differ sharply in cost-effectiveness. The TTT adapter delivers most of the benefit (mAP 0.9605 $\rightarrow$ 0.9623) for a $1.4\times$ increase in RTF and no extra FLOPs. TEA, in contrast, adds only a further +0.0008 mAP at $K = 2$ yet raises RTF by roughly $4\times$ and FLOPs sevenfold (Table 6), because each adaptation step requires a full forward–backward pass. As a result, the complete system runs at about the same speed as the LSTM-based FDE-RNN baseline, largely forfeiting the efficiency advantage of the Mamba backbone. For deployments where inference cost matters, the TTT adapter alone is the more sensible operating point; TEA is justified only when the small additional mAP is worth a severalfold latency increase. We report this plainly rather than presenting the best-mAP system as unconditionally preferable.

Sensitivity of TEA. TEA is also delicate to configure. Tables 4 and 5 show that performance collapses if the learning rate is raised beyond 10^{-3} or if more than a few adaptation steps are taken—both symptoms of over-adaptation, in which the embedding drifts away from a useful speaker representation when pushed too hard by the self-supervised objective. The method is therefore beneficial only as a small, tightly bounded correction. This fragility is itself a finding: unconstrained test-time adaptation of the speaker embedding is not robust for PVAD, and careful regularization of the adaptation step is necessary.

Why the gains are modest on the clean protocol. The gains from test-time adaptation on the standard LibriSpeech test partition, while consistent, are small. We attribute this to the fact that this protocol presents limited distribution shift for either mechanism to absorb. On the acoustic side, although multi-condition training adds reverberation and noise, the train and test distributions are drawn from the same corpus and augmentation pipeline, so the train–test acoustic mismatch that the TTT adapter is designed to correct is mild; on the speaker side, the enrollment embeddings are generally well matched to the test condition, limiting what TEA can recover. To probe whether the mechanisms would indeed provide larger benefits once mismatch is present, we conduct a controlled distribution-shift analysis in Section 5.7. The results there show that the mAP gain grows monotonically with the induced mismatch, reaching +0.033 under strong enrollment mismatch and +0.020 under strong acoustic mismatch—respectively, $9.2\times$ and $5.5\times$ the clean-test gain—and that a coherent precision-oriented adaptation signature emerges across four independent metrics.

Limitations. Three limitations follow. First, although we verified that the proposed system is stable across repeated evaluations of a fixed trained model (Section 5.1), each model configuration was trained only once; given the small effect sizes on the clean test partition, repeating the *training* across several random seeds and reporting the resulting variance would further strengthen the conclusions. We therefore regard the clean-test gains as suggestive rather than definitive, while noting that the controlled distribution-shift analysis of Section 5.7

provides convergent evidence from an independent axis—the coherent dose–response of four metrics across two orthogonal perturbation axes is not readily explained by training-run randomness. Second, the benefit of TEA is contingent on the TTT adapter and does not appear in isolation, so its standalone value remains unestablished. Third, the LibriSpeech-based protocol uses concatenated read speech with artificially clean speaker boundaries, well-matched enrollment, and only synthetic acoustic augmentation; evaluation under genuine overlapping speech, real acoustic-environment mismatch, and mismatched enrollment remains important future work. Our controlled synthetic perturbations of Section 5.7 are a first step toward that setting; they do not replace evaluation on genuine mismatched corpora, which we regard as the natural next step for adaptive PVAD.

5.7. Distribution Shift Analysis

The results reported so far are obtained on the standard LibriSpeech PVAD protocol, whose train and test distributions are closely matched. Because test-time adaptation is motivated precisely by the presence of distribution shift (Section 2), a natural question is how the observed gains scale with the amount of mismatch actually present at inference time. This subsection reports that analysis. We emphasize that the analysis is a controlled probe—test-time perturbations of otherwise unchanged data—and not a substitute for evaluation on genuinely mismatched corpora, which we identify as future work in Section 5.6.

Design. We fix the trained model (the FDE-Mamba backbone that produces the baseline row of Table 2) and, at inference time only, inject zero-mean Gaussian noise into one of two targets. In the *acoustic-perturbation* sweep, the noise is added to the 40-dimensional log-Mel feature stream $\mathbf{x}_{1:T}$ before it enters the backbone; the standard deviation σ_{ac} controls the perturbation magnitude. In the *embedding-perturbation* sweep, the noise is added to the 256-dimensional target speaker d-vector $\mathbf{e}$ before it enters the FiLM generator, with standard deviation σ_{em} . In both cases the same per-batch random seed is reused for the baseline and for the proposed system, so that the noise realization is identical and the gain reflects the effect of adaptation on a matched perturbation. Training data, training procedure, and post hoc Gaussian smoothing ($\sigma = 5$) are unchanged; only the inference-time input is perturbed. Throughout this section, “clean” and the $\sigma = 0$ row denote the *unperturbed* inference condition, in which no additional noise is injected; the underlying multi-condition training augmentation (reverberation and additive noise, Section 4) is retained at every perturbation level, including $\sigma = 0$. The sweep therefore measures shift induced *on top of* an already augmentation-hardened model rather than a clean-versus-noisy contrast.

We interpret σ_{ac} as a proxy for train–test acoustic mismatch—unseen recording channels or environments would perturb $\mathbf{x}_{1:T}$ from the training distribution in ways that Gaussian noise abstracts to a single controllable magnitude—and σ_{em} as a proxy for enrollment mismatch: an enrollment recorded from a poorly matched session, channel, or utterance would displace the d-vector from where it would sit under clean enrollment. These abstractions are deliberately simple; their purpose is to expose the trajectory of the gain as a function of mismatch magnitude, not to reproduce any particular real-world nuisance condition.

Embedding-perturbation results. Table 9 reports four metrics—accuracy, recall, precision, and mAP—of the FDE-Mamba baseline and of the proposed system across seven values of σ_{em} from 0 (matched enrollment) to 0.08 (severely displaced d-vector).

From Table 9, four observations stand out:

1. The mAP gain increases monotonically from +0.0036 at $\sigma_{em} = 0$ to +0.0332 at $\sigma_{em} = 0.08$ —a $9.2\times$ amplification. This is the key qualitative finding: the value of adaptation scales with the amount of mismatch present, precisely the regime in which test-time adaptation is motivated (Section 2).

2. Accuracy tells the same story. The Accuracy gain grows monotonically from +0.32 p.p. to +0.84 p.p. ($2.6\times$), even as the absolute accuracy of both systems collapses under strong mismatch.
3. Precision gains likewise expand overall, from +0.018 at $\sigma_{em} = 0$ to +0.066 at $\sigma_{em} = 0.08$ ($3.7\times$), with small fluctuations at low perturbation levels rather than a strictly monotonic climb.
4. Recall moves in the opposite direction: the proposed system undershoots the baseline by -0.014 at $\sigma_{em} = 0$ and by -0.186 at $\sigma_{em} = 0.08$. This is not a failure mode; it is the recall–precision trade-off already identified in Section 5.6 (“the adapter makes the model more conservative in declaring a frame to be target-speaker speech”), and the fact that it deepens under stronger mismatch shows that the adaptation mechanisms respond to mismatch with a coherent, precision-oriented shift in operating point rather than with uncorrelated noise.

Taken together, the four metrics move in mutually consistent directions across the seven perturbation levels: the mAP and accuracy gains increase monotonically, the precision gain expands overall, and the recall gap widens overall, a signature of a single underlying mechanism—“adaptation trades recall for precision, and does so more strongly the more mismatched the enrollment”—rather than of independent metric noise.

Table 9. Distribution-shift analysis on the enrollment axis: Gaussian noise of standard deviation σ_{em} is added to the target speaker d-vector at inference time. Baseline is the FDE-Mamba backbone; Proposed is the full system (TTT + TEA + Gauss). Δ columns report Proposed – Baseline. All quantities include Gaussian smoothing ($\sigma = 5$).

σ_{em}	Accuracy (%)			Base	Recall		Base	Precision			Base	mAP	
	Base	Prop.	Δ		Prop.	Δ		Prop.	Δ	Prop.		Δ	
0.000	89.87	90.19	+0.32	0.870	0.856	−0.014	0.881	0.899	+0.018	0.9605	0.9641	+0.0036	
0.005	89.69	90.06	+0.37	0.868	0.857	−0.011	0.881	0.899	+0.018	0.9537	0.9633	+0.0096	
0.010	88.96	89.41	+0.45	0.862	0.849	−0.013	0.890	0.901	+0.011	0.9487	0.9615	+0.0128	
0.020	88.08	88.65	+0.57	0.836	0.830	−0.006	0.892	0.904	+0.012	0.9313	0.9462	+0.0149	
0.030	85.96	86.58	+0.62	0.795	0.734	−0.061	0.880	0.899	+0.019	0.9138	0.9312	+0.0174	
0.050	80.64	81.33	+0.69	0.700	0.529	−0.171	0.802	0.846	+0.044	0.8331	0.8548	+0.0217	
0.080	73.02	73.86	+0.84	0.568	0.382	−0.186	0.716	0.782	+0.066	0.7303	0.7635	+0.0332	

Acoustic-perturbation results. Table 10 reports the same four metrics under acoustic perturbation. The picture is qualitatively identical: the mAP gain grows monotonically, and accuracy, precision, and the recall gap all expand overall as σ_{ac} increases.

Table 10. Distribution-shift analysis on the acoustic axis: Gaussian noise of standard deviation σ_{ac} is added to the log-Mel feature stream at inference time. Same conventions as Table 9.

σ_{ac}	Accuracy (%)			Base	Recall Prop.	Δ	Precision			Base	mAP	
	Base	Prop.	Δ				Prop.	Δ	Prop.		Δ	
0.0	89.87	90.19	+0.32	0.870	0.856	−0.014	0.881	0.899	+0.018	0.9605	0.9641	+0.0036
0.1	89.85	90.19	+0.34	0.868	0.857	−0.011	0.878	0.899	+0.021	0.9602	0.9642	+0.0040
0.2	89.76	90.08	+0.32	0.860	0.849	−0.011	0.872	0.901	+0.029	0.9597	0.9641	+0.0044
0.3	89.39	89.83	+0.44	0.843	0.830	−0.013	0.868	0.904	+0.036	0.9571	0.9619	+0.0048
0.5	87.52	88.05	+0.53	0.770	0.734	−0.036	0.871	0.900	+0.030	0.9307	0.9359	+0.0052
0.8	78.76	79.36	+0.60	0.551	0.509	−0.042	0.843	0.866	+0.023	0.8193	0.8363	+0.0170
1.0	72.42	73.17	+0.75	0.429	0.382	−0.047	0.775	0.807	+0.032	0.7401	0.7599	+0.0198

The mAP gain grows monotonically from +0.0036 at $\sigma_{ac} = 0$ to +0.0198 at $\sigma_{ac} = 1.0$, a $5.5\times$ amplification. Accuracy and precision gains expand overall along the axis, and the recall gap widens overall, mirroring the enrollment-axis result.

Interpretation across both axes. Across the two orthogonal perturbation axes and four metrics, we thus observe eight dose–response trajectories. The two mAP trajectories are strictly monotonic across all seven perturbation levels on their respective axes; the remaining six trajectories follow the same overall direction predicted by the “adaptation trades recall for precision under mismatch” interpretation, with only small local fluctuations at low perturbation levels, and none reverses its overall trend. This coherence provides two mutually reinforcing conclusions.

Concerning the mechanism, the systematic sensitivity of the four metrics to the induced mismatch—and their consistent sign patterns—indicates that the adaptation mechanisms are actively responding to distributional shift, rather than producing an essentially random offset from the baseline. If the gains reflected training-run randomness or measurement noise, the trajectories would not track the perturbation magnitude this systematically nor cohere with a single underlying trade-off. As an illustrative measure of how unlikely this coherence is under a random-offset null: if the seven per-level mAP gains on an axis were random offsets unrelated to the perturbation, they would be equally likely to appear in any order, so their falling in exactly the predicted strictly increasing order has probability $1/7! \approx 2 \times 10^{-4}$ per axis, and $(1/7!)^2 \approx 4 \times 10^{-8}$ for both axes jointly—an approximate estimate rather than a formal test, since gains at successive levels share test data and are not independent draws.

Concerning the magnitude of the improvements, the fact that the mAP, accuracy, and precision gains under strong mismatch are several times larger than on the clean protocol reconciles the “consistent but modest” framing of Sections 5.1 and 5.2 with the motivation of test-time adaptation: the clean LibriSpeech protocol simply does not exercise the mismatch regime in which adaptation is most useful, and once mismatch is introduced in a controlled way, the benefits scale with it. We caution, however, that our perturbations remain synthetic; they establish a mismatch-versus-gain trajectory but do not substitute for evaluation under real acoustic-environment or enrollment mismatch, which we identify as the primary future direction (Section 5.6).

6. Conclusions and Future Work

We presented an empirical study of test-time adaptation for personal voice activity detection, introducing a VAD-gated Test-Time Training (TTT) adapter and a test-time speaker embedding adaptation scheme (TEA) on top of the FDE-Mamba backbone. To our knowledge, this is the first application of TTT to PVAD, and the first to adapt the target speaker embedding itself at inference time for this task. Across both an LSTM-based and a selective state-space backbone, the two mechanisms produce consistent but modest improvements in mAP and precision, concentrated on speaker-discrimination metrics rather than overall accuracy, at the cost of a small reduction in recall. Separately from these adaptation results, we identified that a model-agnostic Gaussian smoothing step accounts for the largest single accuracy gain on the FDE-Mamba backbone, and that this gain is unrelated to the adaptation mechanisms and should not be conflated with them.

Taken together, these findings provide a measured baseline for adaptive PVAD and clarify the cost–benefit profile of different adaptation choices: the VAD-gated TTT adapter offers low-cost, inference-only gains with minimal architectural changes, while TEA yields additional but smaller improvements at a noticeable increase in computational cost and latency. We also highlighted several limitations of the present study, including the reliance on a single training seed per configuration, the use of a LibriSpeech-based protocol that does

not explicitly enforce severe acoustic or enrollment mismatch, and the fact that TEA's benefit appears most clearly when combined with TTT rather than as a standalone mechanism.

These limitations point directly to several directions for future work. First, extending the evaluation to more challenging and realistic scenarios—such as far-field capture, reverberant and noisy environments, heavily mismatched or short enrollments, and overlapping multi-speaker speech—would better stress the conditions under which test-time adaptation is most likely to help and may reveal larger gains than those observed here. Second, conducting multi-seed training and broader hyperparameter sweeps would allow a more rigorous assessment of the statistical significance and robustness of the reported improvements. Third, it would be valuable to explore alternative adaptation targets and schedules, including cross-utterance or continual test-time adaptation for PVAD, while explicitly addressing stability and catastrophic forgetting, as well as integrating TTT and TEA with stronger pretraining or self-supervised front-ends. We hope that the empirical baseline and analysis provided in this work will serve as a reference point for future research on adaptive PVAD and, more broadly, on test-time adaptation for speaker-conditioned streaming speech tasks.

Author Contributions: Conceptualization, T.-Y.C., C.-C.C. and J.-W.H.; methodology, T.-Y.C., C.-C.C. and J.-W.H.; software, T.-Y.C. and C.-C.C.; validation, T.-Y.C., C.-C.C., J.-S.L. and J.-W.H.; formal analysis, J.-S.L. and J.-W.H.; investigation, J.-W.H.; resources, J.-S.L. and J.-W.H.; data curation, T.-Y.C., C.-C.C. and J.-W.H.; writing—original draft preparation, J.-W.H.; writing—review and editing, J.-S.L. and J.-W.H.; visualization, J.-W.H.; supervision, J.-W.H.; project administration, J.-W.H.; funding acquisition, J.-W.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Publicly available datasets were analyzed in this study: LibriSpeech (<https://www.openslr.org/12> (accessed on 12 July 2026)) and MUSAN (<https://www.openslr.org/17> (accessed on 12 July 2026)).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AP	Average Precision
ASR	Automatic Speech Recognition
EMA	Exponential Moving Average
FDE	Flexible Dynamic Encoder
FiLM	Feature-wise Linear Modulation
FLOPs	Floating-Point Operations
GPU	Graphics Processing Unit
LSTM	Long Short-Term Memory
mAP	mean Average Precision
ns	non-speech
ntss	non-target speaker speech
PVAD	Personal Voice Activity Detection
RNN	Recurrent Neural Network
RTF	Real-Time Factor
SSM	State Space Model
TEA	Test-time Embedding Adaptation
TENT	Test entropy minimization
tss	target speaker speech
TTA	Test-Time Adaptation
TTT	Test-Time Training
VAD	Voice Activity Detection

References

1. Eyben, F.; Wengier, F.; Squartini, S.; Schuller, B. Real-life voice activity detection with LSTM Recurrent Neural Networks and an application to Hollywood movies. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*; IEEE: Piscataway, NJ, USA, 2013; pp. 483–487. [\[CrossRef\]](#)
2. Jia, F.; Majumdar, S.; Ginsburg, B. MarbleNet: Deep 1D Time-Channel Separable Convolutional Neural Network for Voice Activity Detection. In *Proceedings of the ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: Piscataway, NJ, USA, 2021; pp. 6818–6822. [\[CrossRef\]](#)
3. Ding, S.; Wang, Q.; Chang, S.Y.; Wan, L.; Moreno, I.L. Personal VAD: Speaker-Conditioned Voice Activity Detection. In *Proceedings of the Odyssey: The Speaker and Language Recognition Workshop*; International Speech Communication Association (ISCA): Baixas, France, 2020; pp. 433–439. [\[CrossRef\]](#)
4. Ding, S.; Rikhye, R.; Liang, Q.; He, Y.; Wang, Q.; Narayanan, A.; O'Malley, T.; McGraw, I. Personal VAD 2.0: Optimizing Personal Voice Activity Detection for On-Device Speech Recognition. In *Proceedings of the Interspeech 2022*; International Speech Communication Association (ISCA): Baixas, France, 2022; pp. 3744–3748. [\[CrossRef\]](#)
5. Perez, E.; Strub, F.; de Vries, H.; Dumoulin, V.; Courville, A. FiLM: Visual Reasoning with a General Conditioning Layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*; Association for the Advancement of Artificial Intelligence (AAAI): Palo Alto, CA, USA, 2018; pp. 3942–3951. [\[CrossRef\]](#)
6. Liu, F.; Xiong, F.; Hao, Y.; Zhou, K.; Zhang, C.; Feng, J. AS-pVAD: A Frame-Wise Personalized Voice Activity Detection Network with Attentive Score Loss. In *Proceedings of the ICASSP 2024–2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: Piscataway, NJ, USA, 2024; pp. 11476–11480. [\[CrossRef\]](#)
7. Zeng, B.; Cheng, M.; Tian, Y.; Liu, H.; Li, M. Efficient Personal Voice Activity Detection with Wake Word Reference Speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: Piscataway, NJ, USA, 2024; pp. 12241–12245. [\[CrossRef\]](#)
8. Yu, E.L.; Wang, C.C.; Hung, J.W.; Huang, S.C.; Chen, B. Flexible VAD-PVAD Transition: A Detachable PVAD Module for Dynamic Encoder RNN VAD. In *Proceedings of the Interspeech*; International Speech Communication Association (ISCA): Baixas, France, 2025; pp. 5793–5797. [\[CrossRef\]](#)
9. Chiu, C.C.; Chen, T.Y.; Wang, T.W.; Chen, B.; Hung, J.W. FDE-Mamba: Selective State Space Modeling for Personal Voice Activity Detection. *Appl. Sci.* **2026**, *16*, 4688. [\[CrossRef\]](#)
10. Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B.; Darrell, T. Tent: Fully Test-Time Adaptation by Entropy Minimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Virtual, 3–7 May 2021.
11. Shen, H.; Jiang, W.; Huang, J. Speaker Adaptation for Lip Reading with Robust Entropy Minimization and Adaptive Pseudo Labels. In *Proceedings of the International Conference on Computing, Machine Learning and Data Science*, New York, NY, USA, 23–25 August 2024. [\[CrossRef\]](#)
12. Sun, Y.; Wang, X.; Liu, Z.; Miller, J.; Efros, A.A.; Hardt, M. Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. In *Proceedings of the International Conference on Machine Learning (ICML)*; PMLR: Brookline, MA, USA, 2020; pp. 9229–9248.
13. Sun, Y.; Li, X.; Dalal, K.; Xu, J.; Vikram, A.; Zhang, G.; Dubois, Y.; Chen, X.; Wang, X.; Koyejo, S.; et al. Learning to (Learn at Test Time): RNNs with Expressive Hidden States. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*; PMLR: Brookline, MA, USA, 2025; Volume 267, pp. 57503–57522.
14. Behera, A.; Easow, R.A.; Parvathala, V.; Murty, K.S.R. Test-Time Training for Speech Enhancement. In *Proceedings of the Interspeech 2025*; International Speech Communication Association (ISCA): Baixas, France, 2025; pp. 2375–2379. [\[CrossRef\]](#)
15. Wan, L.; Wang, Q.; Papir, A.; Moreno, I.L. Generalized End-to-End Loss for Speaker Verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: Piscataway, NJ, USA, 2018; pp. 4879–4883. [\[CrossRef\]](#)
16. Yu, E.L.; Chang, R.X.; Hung, J.W.; Huang, S.C.; Chen, B. COIN-AT-PVAD: A Conditional Intermediate Attention PVAD. In *Proceedings of the Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–5.
17. Makishima, N.; Ihori, M.; Tanaka, T.; Takashima, A.; Orihashi, S.; Masumura, R. Enrollment-Less Training for Personalized Voice Activity Detection. In *Proceedings of the Interspeech*; International Speech Communication Association (ISCA): Baixas, France, 2021; pp. 346–350. [\[CrossRef\]](#)
18. Yu, E.L.; Ho, K.H.; Hung, J.W.; Huang, S.C.; Chen, B. Speaker Conditional Sinc-Extractor for Personal VAD. In *Proceedings of the Interspeech*; International Speech Communication Association (ISCA): Baixas, France, 2024; pp. 2115–2119.
19. Gu, A.; Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *Proceedings of the Conference on Language Modeling (COLM)*; Springer: Cham, Switzerland, 2024. [\[CrossRef\]](#)

20. Bovbjerg, H.S.; Jensen, J.; Østergaard, J.; Tan, Z.H. Self-Supervised Pretraining for Robust Personalized Voice Activity Detection in Adverse Conditions. In *Proceedings of the ICASSP 2024–2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: Piscataway, NJ, USA, 2024; pp. 10126–10130. [[CrossRef](#)]
21. Dong, J.; Jia, H.; Chatterjee, S.; Ghosh, A.; Bailey, J.; Dang, T. E-BATS: Efficient Backpropagation-Free Test-Time Adaptation for Speech Foundation Models. In *Proceedings of the The Thirty-ninth Annual Conference on Neural Information Processing Systems*, San Diego, CA, USA, 9–15 December 2025.
22. Dumpala, S.H.; Sastry, C.S.; Oore, S. Test-Time Training for Speech. In *Proceedings of the Workshop on Efficient Systems for Foundation Models (ES-FoMo) at the International Conference on Machine Learning (ICML)*, Honolulu, HI, USA, 23–29 July 2023.
23. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. LibriSpeech: An ASR Corpus Based on Public Domain Audio Books. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: Piscataway, NJ, USA, 2015; pp. 5206–5210. [[CrossRef](#)]
24. Snyder, D.; Chen, G.; Povey, D. MUSAN: A Music, Speech, and Noise Corpus. *arXiv* **2015**, arXiv:1510.08484. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.